Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5559--96-7823

Real-Time Super-High-Speed-Processing

LYNN M. KOFFLEY

Transmission Technology Branch
Information Technology Division

March 13, 1996

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE March 13, 1996	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE Real-Time Super-High-Speed Processing	5. FUNDING NUMBERS PE - 63013N WU - 55-2720-0-5
6. AUTHOR(S) L.M. Koffley	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Washington, DC 20375-5320	8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/5559--96-7823
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Navy Engineering Logistics Office Arlington, VA 22215-5000	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Recent technology advances have greatly improved the ease with which computationally-intense time-critical applications can be implemented in hardware and processed at very high speeds The numerous benefits of implementing algorithms in hardware, and additional benefits of new hardware technology advances in small supercomputing devices are discussed in detail. Case examples showing comparisons between the computational speed of conventional microprocessors versus hardware implementations built around field-programmable gate arrays are analyzed. Several applications that have benefited from these new technology advances are discussed.

14. SUBJECT TERMS		15. NUMBER OF PAGES 15
Real-time Supercomputing Parallel processing	Field-programmable gate array High-speed processing High-speed data throughput	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

Real-Time Super-High-Speed-Processing

I. BACKGROUND

It is well-known that logic (algorithms) implemented in hardware are computationally faster than logic (algorithms) implemented in software. Therefore, most basic, well-defined computational processes in a computer are implemented in hardware. Since not all applications can be completely defined, computers employ processors that allow software to be developed for custom applications.

Software, even low-level software, requires memory accesses for data and instructions, execution of the instructions, and storage of computational results. Typically, instructions take 1-3 processor clock cycles to execute. Even though logic employed on the processor board may finish executing in less than one clock cycle, computational results cannot be stored, displayed, or output until the following clock cycle. All data and addresses must be clocked in order to ensure synchronization.

Reduced Instruction Set Chip (RISC) technology was developed to provide processors capable of executing particular functions in less clock cycles, and to implement additional computational functions in hardware. These processors are often used in dedicated hardware and custom hardware designs. Although usually more difficult to program, RISC processors offer higher computational speed for many custom applications.

Many applications that are well-defined, however, can be implemented directly in hardware, overcoming the many timing limitations of microprocessors. The recent advances in supercomputing, parallel processing, and field programmable gate arrays (FPGAs) have greatly increased the ease with which a designer can implement logic directly in hardware.

II. ALGORITHMIC IMPLEMENTATIONS IN HARDWARE

Implementations of logic and algorithms in FPGAs (such as Xilinx and Altera chips) can clock data through in nanoseconds, compared to the microseconds required by microprocessors. In addition, data paths and feedback loops can be designed to provide

Manuscript approved December 28, 1995.

data inputs at the appropriate time to eliminate additional time requirements for data accesses. When the algorithms are implemented by the programmable logic array, time required for instruction accesses is also eliminated. Therefore, typical instructions requiring 3 clock-cycles to execute at 10 microseconds per clock cycle (30 microseconds), would typically take 10 nanoseconds to execute in field programmable gate array logic. This equates to a 3000 times improvement in execution speed!

Many applications emerging within the last 17 years require enormous processing capability and/or require real-time processing capability. Application areas include cryptography, network scheduling, communications routing, network analysis, image processing, image recognition, real-time control (especially higher-order, non-linear, and stochastic), filtering, video compression, speech recognition, robotics, and artificial intelligence.

Recent technology has increased the ease with which these applications can be implemented in field programmable gate array logic. Commercial Off-The-Shelf (COTS) boards with various FPGA configurations (varying the width and depth of the array) are available, allowing the processing of various lengths of data through various cycles, or levels, of processing. COTS boards are available for plugging into various standard buses, allowing for interfaces to computers, test equipment, and even specialized hardware designed around standardized buses.

III. APPLICATION-SPECIFIC DEVELOPMENT ADVANCES

Since COTS products are available for interfacing to standard platforms, logic design and development can take place on standard platforms and be quickly downloaded to the COTS board. The algorithmic implementation can then be executed on the COTS board within the computer to provide an easy data path for processing information, or can be easily integrated into equipment or a portable computer designed around a standardized bus.

The ease with which programmable logic arrays can be programmed has greatly increased, to the point that algorithms can be defined either by program logic or by block diagrams. Software developed by some companies (i.e., Data I/O) will produce the programmable gate array logic design based on this program code or block diagram. This eliminates the need for the programmer, or designer, to perform detailed low-level logic design. In addition, companies (e.g., Supercom Systems Company) are available that provide engineering services for taking adequate computational descriptions in virtually any form and implementing the design/description into programmable logic arrays. This can provide a fast, inexpensive alternative to writing complex programs, with a resulting design that is orders of magnitude more computationally-efficient.

Many computationally-intense applications require enormous amounts of memory for processing. Implementations in programmable gate array logic can significantly reduce the memory requirements of many of these applications, since data can be processed by the programmable gate array logic until final results are obtained, eliminating the need to store mounds of intermediate results. Also, in the case of many time-critical applications (e.g., network analysis), data can be processed as it arrives, eliminating the need for deep buffers.

Many computationally-intense applications require not only fast processing of algorithms but also fast I/O (input/output of data). COTS products have been built that plug into several standard busses and have an additional, high-speed interface for fast I/O. For instance, Supercomputer Research Corporation (SRC) and Supercom Systems Company have both built COTS boards that supply an additional, programmable external interface that can be clocked externally to provide data rates of 150-260 MBytes/sec.

IV. RECENT APPLICATIONS OF SUPER-HIGH-SPEED TECHNOLOGY

Supercomputer Research Corporation has developed a processor-array board that has been used for a number of applications that require extensive computational capability, including video image processing (with filtering, image enhancement, edge detection, and region detection), video compression (to reduce hardware costs and to reduce processing time by 1 to 2 orders of magnitude), frequency analysis, fingerprint matching, and database text searching. Additional prime applications include "system modeling of: digital signal processors, network protocols, control systems, encoders/decoders, and format converters".

NB Engineering historically used very expensive conventional computers to do image compression. They switched to Xilinx processors to reduce hardware costs and to reduce the processing time by 2 orders of magnitude. NB Engineering is also using Xilinx technology to design better production line defect detection.

Xilinx processing arrays have also been used as an effective means of implementing database searches. The University of Michigan and others developed algorithms to match fingerprints against a database of known finger prints. Dr. Duncan Buell, at Supercomputer Research, developed applications to do database searches for text strings (words, sentences, paragraphs, and larger).

Xilinx processor arrays can be used to model a number of digital systems and communication networks due to:

- the high-processing speeds
- the large number of interconnects between processing nodes
- the large number of logic elements
- the capability of current array boards

This type of modeling could shorten research and developmental schedules, and reduce development and production costs over many conventional means.

An area being targeted for application of Xilinx technology is optical character recognition (OCR). Current commercial OCR units are considerably slower than optical scanners. If implemented using COTS Xilinx boards programmed for optical character recognition, the character recognition and conversion to text could be done in real-time as fast as the scanner could operate. OCR implementations based on Xilinx technology will probably be seen on the market shortly.

Another targeted area is cryptology. Ross Anderson from the United Kingdom, in commenting on the effectiveness of the GSM A5 algorithm, commented, "2^40 trial encryptions could take weeks on a workstation, but the low gate count of the algorithm means that a Xilinx chip can easily be programmed to do key search, and an A5 cracker might have a few dozen of these running at maybe 2 keys per microsecond each." Indeed, a Xilinx chip clocking each stage in nanoseconds (rather than microseconds on a microprocessor) and processing several keys in parallel, could easily reduce the required computational time by over two orders of magnitude. That could mean the difference between 10 months of computational time and 24 hours!

V. ADDITIONAL HARDWARE IMPLEMENTATION ADVANTAGES

One of the limiting features of standard application development using microprocessors is the limited bus throughput available with standard buses. Supercomputing processor-array boards have been developed with external interfaces capable of 250 MBytes/sec, or more. For instance, Supercom Systems Corporation has developed an external interface that can be programmed to clock data through as fast as the data can be processed by the on-board Xilinx array.

Implementations of algorithms/programs in hardware rather than software provides an ideal method of obtaining extensive computational power within a small unit. Benchmarks run on the SRC's initial board gave improvements of 4, 6, 8 and 10 times the speed of a Cray computer. One version of the new board has been developed (by Supercom Systems Company) to interface to the PCI bus. The PCI version is 2 1/2 times the speed of a Cray for most applications (even without using the external interface).

Although these boards are designed to execute a specific (downloadable) program, rather than be a general-purpose computer, they have provisions for downloading new programs with the touch of a key. Implementation of programming functions in field programmable gate arrays (FPGAs), such as Xilinx chips, can be done over standardized buses from many host computers. After the FPGAs have been programmed, data can be sent to the FPGAs for processing over the host computer bus or through an external connection. Supercom Systems Company has both 1-board and 2-board PC configurations, that can use either the PCI bus for data transfer or a high-speed external interface. The PCI bus might allow for up to 50MBytes/sec, while this COTS board can get processing speeds greater than 250 MBytes/sec, interfacing through the external bus of the EISA card.

Not only does the external input provide at least 5 times the throughput of a PCI bus. - It processes (or pre-processes) data before data is stored in memory, potentially solving huge memory limitation problems that could otherwise cause the loss of valuable information. In some real-time, high-speed applications (i.e., high-speed network analysis), it is difficult for a host processor to process data before incoming data causes a memory overflow and data must be discarded without being processed. The Xilinx board is capable of handling data at OC-3 rates and by using the Xilinx board as a preprocessor, it can strip out all unnecessary/processed information, and only send important end-user data to host-processor memory. Flags could also be sent, for instance to indicate software to be executed based on algorithm results.

One added bonus of using the external interface is that it can simplify custom board design. In many instances having a separate dedicated data bus means that custom hardware does not have to be designed around a more complex interface and data I/O does not have to compete with a processor for bus cycles. For example, the PCI bus requires the use of a bridge chip for its complex timing requirements (signals propagate down, are amplified, and devices must listen for the amplified signal). The PCI bus must also share its bus time with the host processor, making the PCI bus a complex, limited interface. Using the external interface, data can simply be clocked through as it becomes available, without interfering with the host processing. The cost of this capability from Supercom Systems Company is only $22K ($7K for the PCI interface card plus $15K for the Array Processor Card which fits in an EISA slot).

Also, one should not rely on the I/O capability of standard computers for real-time capability. One particular deficiency is the limited interrupt capability (only one available for the PCI interface), requiring a mail-box look-up to identify the interrupt, no matter how urgent or routine. A Xilinx array processor/FPGA can provide ample interrupt capability, since it can be used to generate flags. Flags can be used to generate hardware interrupts or software interrupts.

Another considerable advantage of implementation using FPGAs is the great reduction or elimination of the hardware production process. FPGAs can be used to eliminate requirements for building custom hardware in time-critical, computationally-intense, or portable-unit applications. Changes in hardware logic for a custom board design require a redesign of the hardware, generation of new artwork, and re-fabrication. Changes in hardware logic on an FPGA, on the other hand, can be done by changing the program logic and downloading the new program from the host computer.

In summary, there are COTS boards that can plug into standard buses (and therefore many computer platforms), have low power requirements (less than 25W in some cases), and offer the processing capability of several Crays. Not only would the boards provide real-time processing capability for virtually any system, but they would greatly reduce memory requirements and development schedules for many applications. In addition, several of these boards offer external dedicated interfaces capable of extremely high-speed data throughput.

VI. SUPERCOMPUTING HARDWARE VS. CONVENTIONAL-COMPUTING CASE EXAMPLE I

Assume a communication network analysis application where a portable Pentium PC is used to decode protocol information in an ATM network in real-time. Assume a pre-processor strips off the ATM packet headers and sends the ATM packet payloads across the PCI bus to be decoded by the Pentium processor. The average number of comparisons required to decode all protocol layers is estimated based on telecommunications protocol literature. However, regardless of the number of comparisons (or for that matter the nature of the data processing), the following analysis shows the estimated improvement in processing capability of a Xilinx processing array compared to the Pentium processor. Two approaches are evaluated. The first approach (Approach A) assumes the limitation of using the PCI bus for data transfers with only the computational improvement of using a Xilinx processing array. The second approach (Approach B) assumes the use of a high-speed data bus, such as that offered with newer COTS Xilinx boards, in addition to using the Xilinx processing array.

Protocol Decode Processing Using a Pentium Processor:

- Processing Time for Host PC:
  - Average No. of Comparisons/Packet=3x12
  - Number of Packets/10MBytes=208,333
  - Aver. No. of Comparisons/10MByte of Data=7,499,988
  - No. of Instructions/Comparison=5
  - No. of Cycles/Instruction=2
  - Speed of Host Processor(AssumePentiumALR90)=90MHz
  - Effective Rate (80% of clock rate)=72MHz
  - Instruction Processing Time=1042ms
  - Data Throughput for PCI Bus=132MBytes/s

- Data Transfer Time for 1MByte=7.6 ns
- No. of Data Trans. Over PCI Bus=10Mx3=30M
- Processing Time Over PCI Bus=228 ms
- Processing Time for 10MBytes=228ms+1042ms=1270ms
- Processing Speed=7.9MBytes/s

Protocol Decode Processing Approach A: Data Transferred via PCI Bus to Xilinx Processing Array for Data Processing

- Processing Time for Array Processor:
  - Clock Speed=33MHz
  - Bandwidth=4 Bytes
  - Processing Speed=132MBytes/s
  - Processing Time for 10MBytes=76 ms
  - Aver. No. of Bytes Transferred to Host PC=2.5MBytes
  - No. Of Data Transfers Over PCI Bus=625,000
  - Speed of PCI Bus=33MHz
  - Transfer Time=19 ms
  - Processing Time For 10MBytes=76+19=95 ms
  - Processing Speed=105MBytes/s

- Improvement Factor=1270/95=13.4X

Protocol Decode Processing Approach B: Data Transferred via External Interface Directly to Xilinx Processing Array for Data Processing

- Processing Time for Host PC:
- Processing Time (From Above)=1270 ms
- Processing Time for Array Processor w/External Input:
  - Clock Speed=33MHz
  - Bandwidth=9 Bytes
  - Processing Speed=298MBytes/s
  - Processing Time for 10MBytes=33.5 ms
- Improvement Factor=1270/33.5=38.0X

As can be seen, Xilinx processing arrays typically offer a performance improvement of 13-14 times over the processing speed of a Pentium processor (regardless of the application). For the case shown above where the application is protocol decoding, the Pentium processor is only able to process 7.9 megabytes of data per second. The Xilinx processing array is able to process 105 megabytes of data per second for this application when restricted to using the PCI bus. If a high-speed data interface is used instead of the PCI bus, data can be processed at a rate of 298 megabytes per second for this communication network analysis application. In addition, more

Xilinx processing arrays could easily be added in parallel, doubling, tripling, etc. the overall rate.

As stated previously, the host processor is involved in other operations. Even if the host processor's primary function is to process this data (in this case decode protocols), the processor also has a number of housekeeping functions to take care of, including transferring data over the bus, accessing instructions in memory, and storing data results in memory. By considering the amount of time required for the Pentium processor to access and store data and instructions, the amount of time available for processing data can be determined. As more data must be transferred across the PCI bus, less time is available for processing the data. The processing rates (using the Pentium processor) for the network protocol decoding application based on data rates of 155 megabytes/second (OC-3 rates) at 10% capacity and 50% capacity are shown below.

Processing Rate for OC-3 Traffic at 10% Utilization:

- Processing Rate for Host PC:
  - Data Throughput for PCI Bus=132MBytes/s
  - Data Transfer Time for 1MByte=7.6 ns
  - For OC-3 at 10% Capacity:
  - Transfer Time to Memory/s=118 ms
  - Remaining (Processing Time)/s=882 ms
  - No. of Additional Data Trans. Over PCI Bus=Nx2
  - Additional PCI Data Transfer Time => Nx2x7.6 ns
  - Average No. of Comparisons/Packet=3x12
  - Number of Packets/1MBytes=20,833
  - No. of Instructions/Comparison=5
  - No. of Cycles/Instruction=2
  - Speed of Host Processor(AssumePentiumALR90)=90MHz
  - Effective Rate (80% of clock rate)=72MHz
  - Instruction Processing Time=> (3x12x20,833xNx5x2)/72M
  - Nx2x7.6ns + (3x12x20,833xNx5x2)/72M = 882 ms
  - 0.0152N + 0.1042N = 882 ms => N = 7.4 MBytes/s

Processing Rate for OC-3 Traffic at 50% Utilization:

- Processing Rate for Host PC:
  - Data Throughput for PCI Bus=132MBytes/s
  - Data Transfer Time for 1MByte=7.6 ns
  - For OC-3 at 50% Capacity:
  - Transfer Time to Memory/s=589 ms
  - Remaining (Processing Time)/s=411 ms

- No. of Additional Data Trans. Over PCI Bus=Nx2
- Additional PCI Data Transfer Time => Nx2x7.6 ns
- Average No. of Comparisons/Packet=3x12
- Number of Packets/1MBytes=20,833
- No. of Instructions/Comparison=5
- No. of Cycles/Instruction=2
- Speed of Host Processor(AssumePentiumALR90)=90MHz
- Effective Rate (80% of clock rate)=72MHz
- Instruction Processing Time=> (3x12x20,833xNx5x2)/72M
- Nx2x7.6ns + (3x12x20,833xNx5x2)/72M = 411 ms
- 0.0152N + 0.1042N = 411 ms => N = 3.4 MBytes/s

As can be seen, the Pentium processor, accessing data over the PCI bus, is able to process 7.4 megabytes of data per second, when data is being transferred at 15.5 megabytes per second (10% of 155 megabytes per second). Since this processing rate is less than the data transfer rate, data will eventually be overwritten before it is processed. If 1 gigabyte of memory is available for storing data before it is processed, unprocessed data will be overwritten with new data in 126 seconds. Of course if the application relies on processing all incoming data for integrity and reliability, then the Pentium processor is unable to meet the application requirements. If data is arriving at OC-3 rates with 50% channel utilization, unprocessed data will be overwritten in just 14 seconds! In addition, if the host processor is handling other operations, the processing speed for the primary application might be reduced to a fraction of the speed computed above!

Since the Xilinx array processor performs its functions independently of the host processor, the incoming data rate does not affect the speed of the Xilinx processing array. Therefore, the Xilinx processing array can process data at a rate of 298 megabytes per second for this application without regard to the transfer rate. In addition, this rate can be increased by adding more Xilinx processors in parallel. Since data transfer over an external interface does not interfere with host processor functions and PCI bus transfers, preprocessing data with a Xilinx array processor frees up the host processor to perform other functions. In addition, since most of the data can be discarded after being preprocessed by the Xilinx processing array, memory requirements in the host system are greatly reduced.

VII. SUPERCOMPUTING HARDWARE VS. CONVENTIONAL-COMPUTING CASE EXAMPLE II

Another example of computational improvement with the use of FPGAs can be shown with a comparison between an i960 microprocessor and its associated CAM (content addressable memory) and an equivalent implementation using a Xilinx processing array. A programmatic/analytic description of the i960 code is shown below:

Description	Instructions	Clock Cycles
Collect port and MAC address (8 bytes)	24	72-96
Generate hash address (algorithm dependent)	10-60	12-240
Hash table look-up	3-5	15-20
Fetch MAC address from entry table	5	15-20
Compare MAC addresses (6 bytes)	6-10	24-40
Fetch port address if a match	3	9-12
Total clock cycles	51-107	147-428

Supercom Systems Company took an existing implementation of this code used for an i960 CAM to store ATM VPI/VCI (Virtual Path Identifier/Virtual Circuit Identifier) connections and designed a Xilinx array to perform the same function. The Xilinx processing array achieved a 400X improvement over the i960 CAM!

VIII. SUMMARY

In summary, real-time super-high-speed-processing technology is available and can be used to greatly improve performance in many time-critical applications. Hardware and software is available to implement these applications with considerable ease and at a much lower cost than many alternative implementation methods. The combination of super-computing capability, parallel processing capability, and field-programmability make this type of technology extremely valuable to a wide range of applications.

APPENDIX - RELATED TECHNOLOGY VENDORS

Software Vendors:

Data I/O
10525 Willows Road N.E.
P.O. Box 97046
Redmond, Washington 98073-9746

Cadence Design Systems, Inc.
6760 Alexander Bell Drive
Suite 140
Columbia, MD 21046

Exemplar/Mentor Graphics
8500 S.W. Creekside Drive
Building 1
Beaverton, OR 97005

Synopsys
P.O. Box 310
Beaverton, OR 97075-9962

Viewlogic
15200 Shady Grove Road
Suite 350
Rockville, MD 20850

Hardware Vendors:

Annapolis Microsystems
190 Admiral Cochrane Drive
Suite 130
Annapolis, MD 21401

Giga Operations Corp.
2374 Eunice
Berkeley, CA 94708

Metalithic Systems, Inc.
9500 South 500 West
Sandy, UT 84070

SuperCom Systems Company
2906 Stoneybrook Drive
Bowie, MD 20715

Virtual Computer Corp.
6925 Canby Avenue
Reseda, CA 91335